IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 9, September 2021

DOI: 10.17148/IJARCCE.2021.10908

A Model for Improving Image Classification Using Convolutional Neural Network for Emergency Situation Reporting

Dumnamene J.S. Sako[1], Friday E. Onuodu[2], Bartholomew O. Eke[3]

Department of Computer Science, Rivers State University, Port Harcourt, Nigeria[1]

Department of Computer Science, University of Port Harcourt, Nigeria[2,3]

Abstract: Artificial neural networks (ANNs) and Deep Learning have shown great improvement in recognizing and classifying objects and images. Convolutional Neural Networks (CNNs) can be exploited to automatically extract features from images using the hierarchical structure. Specifically, we investigate image classification in the context of every day small scale emergency events, where images associated with tweets during the target emergency events were used to train and test classification approaches. We present a method of classifying images using convolutional neural networks that perform classification in layers in order to determine whether or not the image relates to an emergency event and if the image is incident-related, then which category of the target emergency events it belongs to. Experiments on a home-grown dataset show that these methodologies can classify images into the different classes with an F1 score of 88.12%**.**

Keywords: Image classification, convolutional neural networks, social media, Twitter, Emergency events

I. INTRODUCTION

Over the last few years, Artificial Neural Networks (ANNs) and Deep Learning have achieved state-of-the-art performances that have seen their use become widespread in many domains [7] [10] [19]. They have shown great improvement in recognizing and classifying objects and images and have changed the way we handle data in computational data analytics [13]. CNNs, as generic feature extractors, have been continuously improving the image classification accuracy, avoiding the traditional hand-crafted feature extraction techniques in image classification problems; hand-crafted features are usually low-level features without enough mid-level and high-level information, which hinders the performance [18]. The features learned from CNNs are not designed by human engineers, but from data using a general-purpose learning procedure [9].

Deep architectures with hierarchical frameworks enable the representation of complex concepts with fewer nodes than shallow architectures. Unlike traditional statistical approaches where humans were required to study the data carefully and design useful features, deep learning places the machine in charge of learning useful features (or representations) directly from the data without human intervention. Deep learning typically uses high-capacity neural network models, which are trained by a large number of training samples [3].

With increasingly widespread access to mobile phones and the Internet, in times of crisis, it is increasingly common for the public or people affected by the crisis/event to use social media to broadcast their needs, propagate news, post text messages and photographs(images) requesting assistance, describing the situation on the ground and staying abreast of evolving situations [15]. This huge resource may potentially gather a valuable body of information regarding incidents that differ significantly by type, location, and time. Such availability of content-rich data is extremely valuable for emergency management (EM) personnel as they can take more accurate decisions in emergency situations by having textual and/or visual information of the incident [17].

An aspect of social media in relation to emergency management, which has so far received little attention, is image. Images have the potential to provide new insights on top of the text-derived intelligence in tweets, giving a rich and contextual information stream in crisis situations. For example, images of fires provide an immediate cue to crisis coordinators about an event allowing them to react appropriately. Images provide a less ambiguous insight into a situation compared to subjective textual descriptions. An image can show the size of the fire and also provide clues to environmental conditions such as weather conditions and the potential fuel load in the vicinity [8].

In this study, we investigate if images associated with incidents could be processed directly without relying on the associated text to identify high value images depicting every day small scale events like crimes and civil disorder (which includes shootings, kidnapping, terrorist attack, riot, protest, etc), fire, damaged infrastructure, flood caused by heavy rainfall and traffic accidents.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 9, September 2021

DOI: 10.17148/IJARCCE.2021.10908

The rest of the paper is organized as follows. In Section 2, we give brief introduction of Convolutional Neural Networks (CNNs) for image feature extraction and classification. In Section 3, the proposed model is presented. In Section 4, we validate the superiority of the proposed model on a home-grown dataset. Section 5 provides a review of some related works. Finally, Section 6 gives the conclusion.

II. BACKGROUND

The convolutional neural network (CNN) is a class of deep learning neural networks. A CNN works by extracting features from images. This eliminates the need for manual feature extraction. The features are not trained! They're learned while the network trains on a set of images. This makes deep learning models extremely accurate for computer vision tasks [6]. CNNs have an input layer, and output layer, and hidden layers. The hidden layers usually consist of feature-extraction layers (namely convolutional layers, ReLU layers and pooling layers), and classification layers - one or more fully connected layers to take the higher-order features and produce class probabilities or scores. These layers are fully connected to all of the neurons in the previous layer. Basically, a CNN consists of one or more pairs of convolution and pooling layers and finally ends with fully connected neural networks. The feature-extraction layers have a general repeating pattern of the sequence, consisting of the convolution layer and pooling layer. A typical convolutional network architecture is shown in Figure 1

An image CNN starts with an input image which is broken down into pixels. For a black and white image, those pixels are interpreted as a 2D array (for example, 2x2 pixels). Every pixel has a value between 0 and 255. (Zero is completely black and 255 is completely white. The greyscale exists between those numbers.) Based on that information, the computer can begin to work on the data.

For a color image, this is a 3D array with a blue layer, a green layer, and a red layer. Each one of those colors has its own value between 0 and 255. The color can be found by combining the values in each of the three layers. The mathematical processes involved in the feature extraction and classification of images as expressed by [11] are outlined in equations 1 – 12. An image, in general, can be mathematically represented as a tensor with the following dimensions:

$$dim(image) = (n_H, n_W, n_C) \quad (1)$$

where n_H, n_W, and n_C are the size of the Height, Width and the number of Channels respectively. n_C=3 for RGB (Red, Green and Blue) image, and n_C=1 for grayscale image.

In convention, the filter K is considered to be squared and to have an odd dimension denoted f, which allows each pixel to be centered in the filter and thus consider all the elements around. When operating the convolutional product, the filter/kernel K must have the same number of channels as the image, this way we apply a different filter to each channel. Thus the dimension of the filter is as follows:

$$dim(filter) = (f, f, n_C) \quad (2)$$

The image is then convolved with multiple learned kernels using shared weights. A convolutional layer is parametrized by the number of maps, the size of the maps, and kernel sizes. Each layer has M maps of equal size (Mx, My). A kernel of size (Kx, Ky) is shifted over the valid region of the input image. Each map in layer l is connected to all maps in layer $l-1$. Neurons of a given map share their weights but have different input fields (Zhou et al., 2017)

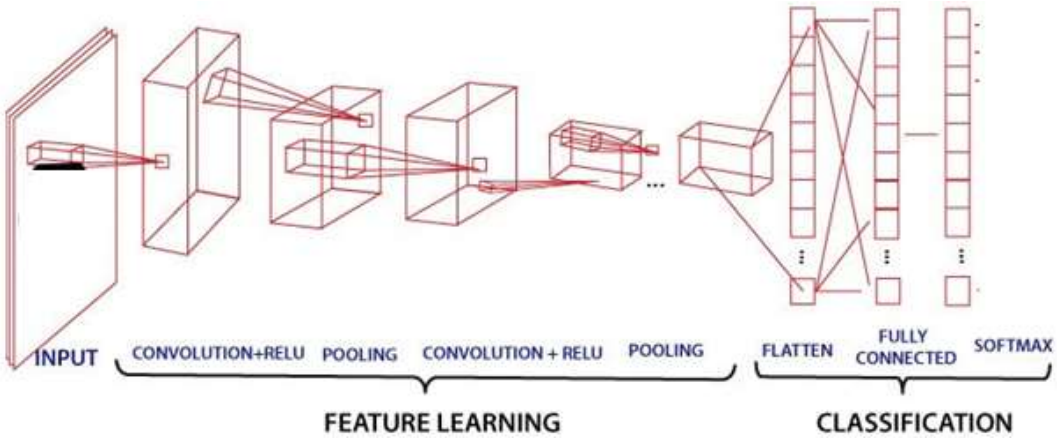

Figure 1: A Typical Convolutional Neural Network Architecture

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 9, September 2021

DOI: 10.17148/IJARCCE.2021.10908

(1) Convolution Layers. The convolution product between the image and the filter is a 2D matrix where each element is the sum of the elementwise multiplication of the cube (filter) and the sub-cube of the given image. Therefore, for a given image and filter we have:

$$conv(I,K)_{x,y} = \sum_{i=1}^{n_H} \sum_{j=1}^{n_W} \sum_{k=1}^{n_C} K_{i,j,k} I_{x+i-1,y+j-1,k} \quad (3)$$

And

$$dim(conv(I,K)) = \left(\left\lfloor \frac{n_H+2p-f}{s} + 1 \right\rfloor, \left\lfloor \frac{n_W+2p-f}{s} + 1 \right\rfloor\right); s > 0 \quad (4)$$
$$= (n_H + 2p - f, n_W + 2p - f); s = 0$$

where $\lfloor x \rfloor$ is the floor function of x.

In a convolution neural network, the f x f x n_C filter's parameters are learned through the backpropagation.

At the convolutional layer, we apply convolutional products, using many filters this time, on the input followed by an activation function ψ.

For the l^{th} convolution layer and given the following notations:

Input: $a^{(l-1)}$ with size $\left(n_H^{(l-1)}, n_W^{(l-1)}, n_C^{(l-1)}\right)$, $a^{(o)}$ being the image in the input; Padding: $p^{(l)}$; Slide: $s^{(l)}$; Number of filters: $n_C^{(l)}$ where each $K^{(n)}$ has the dimension $(f^{(l)}, f^{(l)}, n_C^{(l-1)})$; Bias of the n^{th} convolution: $b_n^{(l)}$; Activation function: $\varphi^{(l)}$; Output: $a^{(l)}$ with size $\left(n_H^{(l)}, n_W^{(l)}, n_C^{(l)}\right)$

$\forall n \in [1,2,\dots,n_C^{(l)}]$:

We have:

$$conv(a^{[l-1]}, K^{(n)})_{x,y} = \psi^{[l]}\left(\sum_{i=1}^{n_H^{[l-1]}} \sum_{j=1}^{n_W^{[l-1]}} \sum_{k=1}^{n_C^{[l-1]}} K_{i,j,k}^{(n)} a_{x+i-1,y+j-1,k}^{[l-1]} + b_n^{[l]}\right) \quad (5)$$
$$dim(conv(a^{[l-1]}, K^{(n)})) = (n_H^{[l]}, n_W^{[l]})$$

Thus:

$$a^{[l]} = \quad (6)$$
$$[\psi^{[l]}(conv(a^{[l-1]}, K^{(1)})), \psi^{[l]}(conv(a^{[l-1]}, K^{(2)})), \dots, \psi^{[l]}(conv(a^{[l-1]}, K^{(n_C^{[l]})}))]$$
$$dim(a^{[l]}) = (n_H^{[l]}, n_W^{[l]}, n_C^{[l]})$$

With:

$$n_{H/W}^{[l]} = \left\lfloor \frac{n_{H/W}^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor; s > 0 \quad (7)$$
$$= n_{H/W}^{[l-1]} + 2p^{[l]} - f^{[l]}; s = 0$$
$$n_C^{[l]} = number\ of\ filters$$

The learned parameters at the l^{th}layer are filters with $\left(f^{(l)}, f^{(l)}, n_C^{(l-1)}\right)$ x $n_C^{(l)}$ parameters, and bias with $(1x1x1)x\ n_C^{(l)}$ parameters (broadcasting)

(2) Pooling: The pooling layer aims at downsampling the features of the input without impacting the number of the channels. The pooling operation is carried out through each channel and thus it only affects the dimensions (n_H, n_W) and keeps n_C intact. The pooling layer has no parameters to learn. Given an image, we slide a filter, with no parameters to learn, following a certain stride, and we apply a function on the selected elements. We have

$$dim(pooling(image)) = \left(\left\lfloor \frac{n_H+2p-f}{s} + 1 \right\rfloor, \left\lfloor \frac{n_W+2p-f}{s} + 1 \right\rfloor, \mathbf{n_C}\right); s > 0 \quad (8)$$
$$= (n_H + 2p - f, n_W + 2p - f, \mathbf{n_C}); s = 0$$

For the l^{th} pooling layer with the following notations:

Input: $a^{[l-1]}$ with size $\left(n_H^{[l-1]}, n_W^{[l-1]}, n_C^{[l-1]}\right)$, $a^{[0]}$ being the image in the input; Padding: $p^{[l]}$; Slide: $s^{[l]}$; Size of the pooling filter: $f^{[l]}$;

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 9, September 2021

DOI: 10.17148/IJARCCE.2021.10908

Pooling function: $\emptyset^{[l]}$; Output: $a^{[l]}$ with size $\left(n_H^{[l]}, n_W^{[l]}, n_C^{[l]} = n_C^{[l-1]}\right)$

We have:

$$a_{x,y,z}^{[l]} = pool(a^{[l-1]})_{x,y,z} = \phi^{[l]}((a_{x+i-1,y+j-1,z}^{[l-1]})_{(i,j)\in[1,2,\dots,f^{[l]}]^2}) \quad (9)$$
$$dim(a^{[l]}) = (n_H^{[l]}, n_W^{[l]}, n_C^{[l]})$$

With:

$$n_{H/W}^{[l]} = \left\lfloor \frac{n_{H/W}^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor ; s > 0 \quad (10)$$
$$= n_{H/W}^{[l-1]} + 2p^{[l]} - f^{[l]}; s = 0$$
$$n_C^{[l]} = n_C^{[l-1]}$$

(3) Fully connected layer - Fully connected layer has full connections to all activations in previous layer to obtain the output that represents high-level features in the data. The two processes of convolutions and pooling, can been thought of as feature extractors. These features are then passed, usually as a reshaped vector of one row, further to the network, to be trained for classification. In a convolutional layer, neurons only receive input from a subarea of the previous layer. In a fully connected layer, each neuron receives input from every element of the previous layer [2]**.** It is a finite number of neurons which takes in input a vector and returns another one. In general, considering the j^{th} node of the i^{th} layer we have the following equations:

$$z_j^{[i]} = \sum_{l=1}^{n_{i-1}} w_{j,l}^{[i]} a_l^{[i-1]} + b_j^{[i]} \quad (11)$$
$$\rightarrow a_j^{[i]} = \psi^{[i]}(z_j^{[i]})$$

The input $a^{[i-1]}$might be the result of a convolution or a pooling layer with the dimensions $\left(n_H^{[i-1]}, n_W^{[i-1]}, n_C^{[i-1]}\right)$.
In order to be able to plug it into the fully connected layer we flatten the tensor to a ID vector having the dimension

$\left(n_H^{[i-1]} \text{ x } n_W^{[i-1]} \text{ x } n_C^{[i-1]}, 1\right)$, thus:

$$n_{i-1} = n_H^{[i-1]}, n_W^{[i-1]}, n_C^{[i-1]} \quad (12)$$

with the learned parameters at the l^{th} layer being the weights with $w_{j,l}$ with n_{l-1} x n_l parameters and bias with n_l parameters.

III. THE PROPOSED MODEL

In this section, we present a CNN-CNN combined model for image classification. The system framework is shown in Figure 2. The proposed model consists of two main parts: Binary classification and multi-class classification. To perform classification for each of the layers of the classification tasks, tweets are captured using Twitter Streaming API, filtered by incident-related keywords. The image content of the tweet is extracted, preprocessed and given as an input to the CNN model for training. The trained model is used for predicting the label of the testing images. The output of the classifier can be predicted by the class label (either incident related tweet or not) in the first instance and if it is incident-related, then which category of target incident/emergency events the image belongs to.

We perform binary classification in the first layer. That is, we aim to classify each message into one of the two classes i.e. “incident-related" vs. “not incident-related". Furthermore, identifying social media messages by category assists Emergency management organizations in coordinating their response. Categories such as crimes and civil disorder, fire, damaged infrastructure, flood caused by heavy rainfall and traffic accidents could therefore be directed to different response agencies. In this work, we show how we can classify tweets into multiple classes in the second layer if the output of the first layer indicates that the image is incident-related.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 9, September 2021

DOI: 10.17148/IJARCCE.2021.10908

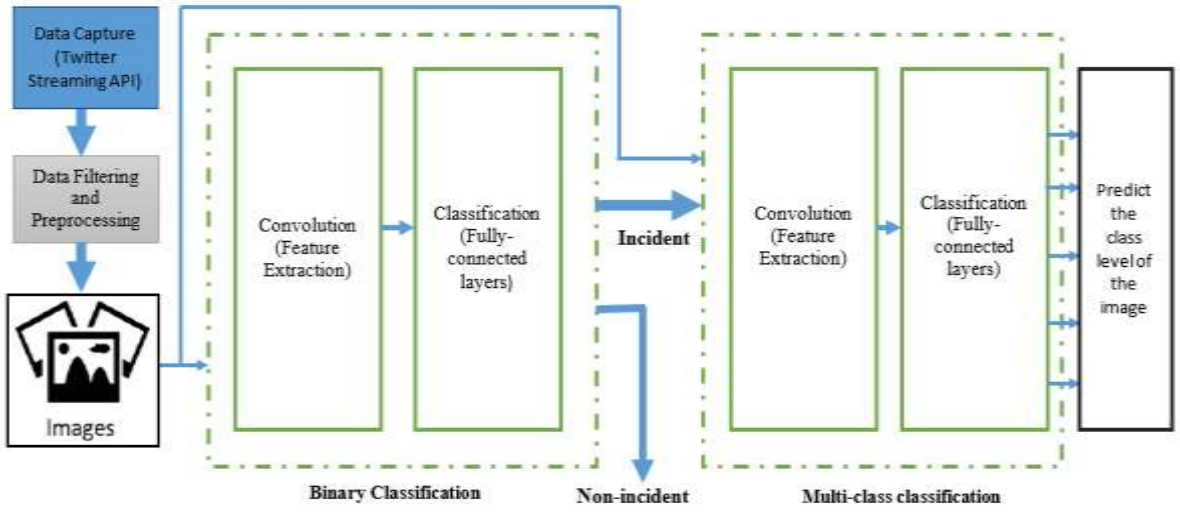

Figure 2: Structure of the Proposed Image CNN Model for classifying incident types of reported target incidents

IV. EXPERIMENTS AND DISCUSSIONS

In this section, we present the experimental results to demonstrate the effectiveness of our proposed CNN model structure for image classification. All the models are built and implemented using Python Programming language based on Keras deep learning library for Python running on tensorflow framework.

4.1 Dataset and Preprocessing

A total of 15483 image samples were collected from (i) Twitter platform using a collection of keywords queries and filter matching for twitter images in tweets using Twitter Streaming API over different time periods to ensure we get a representative sample of crisis related tweets containing images in the target emergency events. (ii) images in human and infrastructural damage dataset [12] and (iii) images from google search using the same keywords. Table 1 shows Description and sizes of the classes in the dataset. Some sample images in the dataset are presented in Figure 3.

We preprocessed the images by reshaping them into the shape the network expects and scaling them so that all values are in the [0,1] interval. For instance, a training data stored in an array of shape (15483, 150,150, 3) of type uint8 with values in the [0,255] interval is transformed into a float32 array of shape (15483,150,150,3) with values between 0 and 1.

TABLE 1 | Description and Sizes of the Classes in the Datasets

Class	**Label**	**Description**
Damaged Infrastructure	1506	Collapsed/Damaged buildings or roads, destroyed bridges, utilities/services interrupted (e.g. falling electricity pole).
Fire	1510	Building fire
Flood	1512	floods caused by heavy rainfall
Traffic Accidents	1550	Motor vehicle crash such as motorcycle accident, car accident, plane crash, etc.
Crime and Civil Disorder	1602	shootings, cultism and terrorism, kidnapping, riot, protest, etc.
Non-incident related	7803	Not useful for emergency response
Total	15483	

4.2 Model Configurations

We experimented with three different CNN models in increasing order of complexity and with varying configurations and the baseline.

The models are summarized as follows:

(i) Baseline model: Support vector machine (SVM) trained and tested to validate the accuracy advantage of our models

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 9, September 2021

DOI: 10.17148/IJARCCE.2021.10908

(ii) CNN_I: without image augmentation and fine tuning
(iii) CNN_{II}: with image augmentation and fine tuning
(iv) VGG16: Pre-trained VGG16 model [16] trained on ImageNet dataset [5], and the weights of the networks' last layers were modified and fine-tuned on our dataset.

4.3 Hyperparameters and Training

We used 80% of the dataset as the training set and 20% as the test set. The validation set is 20% of the training set. We trained a binary classifier and a multi-class classifier. For the binary classifier training, we merge all the incident types to create one general incident-related class. We train the models by for binary and multi-class classifications optimizing the cross entropy using the Adam optimizer for a maximum of 25 epochs. We experimented with {0.25, 0.5} dropout rates, {16, 32} mini-batch sizes and early stopping based on the accuracy on the validation set. We use rectified linear units (ReLU) for the activation function. In all experiments, the models (and hyper parameters) are tuned on the validation (development) set and the final performance evaluated against the test set. The precision, recall, and F_1-score are calculated as the measurements of performance.

(a) Accident (b infrastructural damage (c)fire

(d) flood (e) crime and civil disorder (f) non-incident

Figure 3: Sample Dataset Images

4.4 Results and Discussions

In this section, we present the experimental results and discussions on our classifiers. We evaluate them for the binary and multi-class classification tasks. For the former, we merge all incident types classes to create one general incident-related class.

A. Binary Classification

Table 2 presents the results of binary classification comparing several CNN based classifiers with the SVM classifier. CNN_{II} and the modified VGG16 performed better than the SVM classifier for all events under consideration. With respect to the correctness of classification, the modified VGG16 model achieves the best performance with respect to recall, precision, and F1 with scores of 85.84%, 84.95% and 85.39%, respectively. The best model, VGG16 models perform better than the baseline model by a margin of 5.17%.

B. Multi-class Classification

The precision, recall and F_1 measures of the models on our test dataset for the multi-class image classifications are summarized in Table 3. CNN models performed better than the baseline model. While all the CNN-based models achieved better performance, the VGG16 model outperformed other models on the validation set, and it was also the

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 9, September 2021

DOI: 10.17148/IJARCCE.2021.10908

fastest model for training and prediction. Also, the CNN_I, CNN_{II} and VGG16 models with fine-tuning outperformed the baseline model by 6.40%, 12% and 13.45% respectively.

TABLE 2 | Binary Classification Performance (%).

	SVM			$CNN_I^{(1)}$			$CNN_{II}^{(1)}$			$VGG16^{(1)}$		
Class	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Incident	83.07	78.86	80.91	86.76	78.34	82.34	82.45	74.23	78.12	86.34	85.67	86.00
Non-incident	80.02	79.03	79.52	82.67	72.45	77.22	83.56	81.45	82.49	85.33	84.23	84.78
Average	81.55	78.95	80.22	84.72	75.40	79.78	83.01	77.84	80.31	85.84	84.95	85.39

TABLE 3 | Multi-class Classification Performance (%).

class	**SVM**			$CNN_I^{(2)}$			$CNN_{II}^{(2)}$			$VGG16^{(2)}$		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Accidents	78.87	75.78	80.05	80.34	85.34	82.54	82.56	85.87	87.56	86.78	87.65	87.23
Crimes and Civil Disorder	78.76	72.34	68.65	74.90	85.32	80.34	83.34	82.34	84.39	88.72	89.34	85.89
Damaged Infrastructure	72.12	60.34	67.87	81.34	83.23	82.34	80.15	84.56	86.14	89.23	88.23	89.34
Fire	68.98	78.34	76.88	84.34	79.34	78.90	84.32	83.45	89.90	89.29	87.34	88.87
Flood	70.34	78.98	79.90	78.34	78.23	81.23	80.23	85.34	85.38	85.90	87.09	89.21
Average	73.81	73.16	74.67	79.85	82.29	81.07	82.12	84.31	86.67	87.98	87.93	88.12

V. RELATED WORK

[14] postulated that pictures taken on-the-ground can offer reliable and valuable information for improving situation awareness and could be used as proxy indicators for relevance. They, therefore, established that the presence of images in crisis-related messages or tweets indicates their geographical proximity to the event. They explored various social media platforms, (Twitter, Flickr and Instagram) in the case of floods in Saxony 2013, Germany. From the results obtained, they established that the presence of images in crisis-related messages or tweets indicates their geographical proximity to the event.

[8] proposed a novel application of image classification approaches in the area of Emergency Situation Awareness. They investigated methodologies involving classification of images based on low-level features as well as methods built on top of pre-trained classifiers. Specifically, they investigated image classification in the context of a bush fire emergency where images associated with tweets during emergency were used to train and test classification approaches and showed that the methodologies can classify images into fire and not fire-related classes with an accuracy of 86%.

[4] proposed a methodology to study the occurrence of fires through image posts on Flickr. They collected several years' worth of photos and the spatio-temporal meta-data associated with the photos using fire-related search terms and used an image classification model to detect geotagged photos that are further analysed to determine if a fire event did occur at a particular time and place. However, these approaches are limited to the detection of a single event; floods in [14] and fire in [4, 8].

To address some of the limitations of the approaches within the category of image classification above, [1] presented an end-to-end social media image processing system called Image4Act; a system that can help humanitarian organizations gain situation awareness and to understand the severity of a crisis for better decision-making. The system combined human computation and machine learning techniques to process high-volume social media imagery content in real time during natural and human-made disasters. To cope with the noisy nature of the social media imagery data, they used use deep learning techniques, which are current state-of-art and demonstrate efficient performance, filter out irrelevant and duplicate images.

VI. CONCLUSION

Deep Learning has achieved state-of-the-art performances that have seen their use become widespread in many domains and have shown great improvement in recognizing and classifying objects and images. Convolutional Neural Networks (CNN), a deep learning neural network, has been exploited to automatically extract features from images using the hierarchical structure thereby placing the machine in charge of learning useful features (or representations) directly from the data without human intervention. Specifically, we investigated image classification in the context of every day small

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 9, September 2021
DOI: 10.17148/IJARCCE.2021.10908

scale emergency events, where images associated with tweets during the target emergency events were used to train and test classification approaches

We presented a method of improving image classification using state-of-the-art deep learning convolutional neural networks that perform classification in layers in order to determine whether or not the image relates to an emergency event and if the image is incident-related, then which category of the target emergency events it belongs to. Experiments on a home-grown dataset show that these methodologies can classify images into the different classes with an F1 score of 88.12%.

REFERENCES

[1] Alam, F., Imran, M. and Ofli, F. (2017). Image4Act: Online Social Media Image Processing for Disaster Response. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 601-604. http://dx.doi.org/10.1145/3110025.3110164
[2] Bonner, A. (2019). The Complete Beginner's Guide to Deep Learning: Convolutional Neural Networks and Image Classification. https://towardsdatascience.com/wtf-is-image- classification-8e78a8235acb
[3] Choi, E. (2018). Doctor AI: Interpretable deep learning for modeling electronic health records. A PhD dissertation. Georgia Institute of Technology.
[4] Daly, S. and Thom, J.A. (2016). Mining and Classifying Image Posts on Social Media to Analyse Fires. Long Paper – Social Media Studies Proceedings of the ISCRAM 2016 Conference – Rio de Janeiro, Brazil.
[5] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large- scale hierarchical image database. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 248-255.
[6] Hu, W., Huang, Y., Wei, L., Zhang, F. and Li, H. (2015). Deep convolutional neural networks for hyperspectral image classification. Journal of Sensors, 12.
[7] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. Proceedings of the 26th Annual Conference on Neural InformationProcessing Systems., Lake Tahoe, Nev,. 1097–1105
[8] Lagerstrom, R., Arzhaeva, Y., Szul, P., Obst, O., Power, R., Robinson, B. and Bednarz, T. (2016). Image classification to Support Emergency Situation Awareness. Frontiers in Robotics and AI. https://doi.org/10.3389/frobt.2016.00054
[9] LeCun, Y, Bengio, Y and Hinton, G. (2015). Deep learning, Nature, 521(7553),436–444
[10] Leung, M. K. K., Xiong, H. Y., Lee, L. J. and Frey, B. J.(2014). Deeplearning of the tissue-regulated splicing code, Bioinformatics, 30(12), I121–I129.
[11] Mebsout, I. (2020). Convolutional Neural Networks' Mathematics. https://towardsdatascience.com/convolutional-neural-networks-mathematics-1beb3e6447c0.
[12] Mouzannar, H., Rizk, Y. and Awad, M. (2018). Damage Identification in Social Media Posts using Multimodal Deep Learning. Proceedings of the 15th International Conference on Information Systems for Crisis Response and Management (ISCRAM), Rochester, 529-543.
[13] Nooka, S.P. (2016). Fusion of Mini-Deep Nets. Thesis. Rochester Institute of Technology. https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=10351&context=theses
[14] Peters, R., and Albuquerque, J. P. D. (2015). Investigating images as indicators for relevant social media messages in disaster management, in The 12th International Conference on Information Systems for Crisis Response and Management, Kristiansand, Norway.
[15] Ramchurn, S.D., Trung Dong Huynh, T.D.,Feng Wu, F., Yuki Ikuno, Y., Flann, J., Moreau, L., Fischer, J.E., Jiang, W., Rodden, T., Simpson, E., Reece, S., Roberts, S., Jennings, N. R. (2016). A Disaster Response System based on Human-Agent Collectives. Journal of Artificial Intelligence Research, 57, 661-708
[16] Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[17] Yang, Y., Ha, H.-Y., Fleites, F., Chen, S.-C., and Luis, S. (2011). Hierarchical disaster image classification for situation report enhancement. IEEE 12th International Conference on Information Reuse and Integration (Las Vegas, Nevada).
[18] Yim, J., Ju, J., Jung, H and J. Kim (2015). Image classification using convolutional neural networks with multi-stage feature. Advancesin Intelligent Systems and Computing, 345, 587–594.
[19] Zhou, L., Li, Q., Huo, G., and Zhou, Y. (2017). Image Classification Using Biomimetic Pattern Recognition with Convolutional Neural Networks Features. J. of Computational Intelligence and Neuroscience. https://doi.org/10.1155/2017/3792805

